23 | 存储案例：如何降低SSD峰值延迟?

2020-01-17 庄振运

讲述：庄振运
时长 17:38 大小 14.14M

你好，我是庄振运。

我们之前讲过，存储系统的性能很关键（参见第 17 讲）。我们这一讲就探讨存储方面的优化案例，是关于 SSD 性能的。

现在很多公司里面的高性能存储系统，一般都是基于 SSD 的，这主要归功于 SSD 价格在近几年的大幅度下降。但是，SSD 也不是包治百病的灵丹妙药，也有自己的特殊性能问题。我们今天就重点讲述两点：**SSD 的损耗**和 **IO 访问延迟偶尔过大**的问题。

这里的第二个问题可能听起来很让人吃惊：不是说 SSD 延迟很低吗?

一般情况下，是的。但是特殊情况下就不一定了，这个就说来话长了，它和 SSD 的内部原理有关。我们会一步步地探讨问题形成的原因和解决的策略。

SSD 为什么会损耗?

我们的第 17 讲是关于存储系统的，讲过 SSD 的工作原理和性能。为了防止你忘记，我们就在这里快速地回顾一下其中的一个重要概念：**写入放大**。

什么是写入放大呢？当写入 SSD 的物理数据量，大于应用程序打算写入 SSD 的逻辑数据量时，就会造成“写入放大”。

如果是传统硬盘 HDD，就不会有写入放大的问题。那么 SSD 为什么会有写入放大呢？这是因为 SSD 内部的工作原理和硬盘很不一样。

我们知道，HDD 是可以直接写入覆盖的。和 HDD 不同，SSD 里面的页面只能写入一次，要重写的话，必须先回收擦除，而且只能在“块”这个级别进行擦除。因此呢，SSD 内部就需要不断地移动所存储的数据，来清空需要回收的块。也就是说，SSD 内部需要进行块级别的“垃圾回收”。垃圾收集器必须有效地在 SSD 内部不断地回收块，回收以前使用的页面空间，然后才能在这个块上写入新数据。

因此，对 SSD 的写入需求，比对 HDD 的写入需求更高。

写入放大的缺点是什么？就是**会更快地损耗 SSD 的生命**。

每个 SSD 都有固定数量的擦除周期，如果在很短时间内写到 SSD 太多数据，就会导致 SSD 损耗太快，有可能过早烧坏 SSD。换句话说，很高的写入速率，可能会导致 SSD 在到达其预期使用寿命之前就发生故障。

所以，我们要注意一个常用的指标叫：年损耗率（Burn Rate）。这个指标是怎么定义的呢？是用 SSD 的预期寿命推导出来的。比如一个 SSD 预期寿命是 4 年。那么每年可以损耗 25%，这就是年损耗率。

如何减少 SSD 损耗?

前面讲的“写入放大”，其实也可以用一个相应的具体指标来衡量，就是“**写入放大系数**”；它代表物理写入 SSD 的数据与应用程序写入的逻辑数据之比。比如，如果写入放大系数是 2，就表示写入每 10KB 的逻辑数据，SSD 实际上写了 20KB。为了控制 SSD 的年损耗率，我们需要尽量降低写入放大系数。

那么如何减少写入放大系数呢？常见的方法有两种：

1. 是**保留一定的空闲存储空间**，这是因为写入放大系数是和 SSD 存储空闲率相关的。
2. 是**使用 Trim**。

这两种方法可以同时使用，我们下面分别介绍。

我们先简单说一下第一种方法。每个 SSD 都有一定数量的预留空间，这个空间不是 SSD 可用容量的一部分。这样做是有原因的。尽管我们可以使用工具来调整 SSD 卡上的可用容量，但是我不建议你减少预配置的可用空间，因为这将降低写入性能，并可能大大缩减 SSD 的使用寿命。

我们在存储数据到 SSD 时候，也不要存得太满，也就是不要追求太高的空间使用率。那么我们将 SSD 可用存储容量的使用率目标定为多少比较合适呢？一般来说，我们可以定为 80%至 85%，以保持较低的写入放大率。

SSD 的空闲可用空间越多，内部垃圾收集的开销就越低，就越有可能降低写入放大系数。但是这种关系不是线性的，所以存在着收益递减的问题。

第二种方法是用 Trim。我首先为你讲解一下什么是 Trim。

Trim 是个命令，是操作系统发给 SSD 控制器的特殊命令。使用 Trim 命令，操作系统可以通知 SSD 某些页面存储的数据不再有效了。比如，对于文件删除操作，操作系统会将文件的扇区标记为空闲，以容纳新数据，然后就可以将 Trim 命令发送到 SSD。

Trim 命令有什么好处呢？

SSD 收到 Trim 命令后，SSD 内部的控制器会更新其内部数据页面地图，以便在写入新数据时不去保留无效页面。并且，在垃圾回收期间不会复制无效页面，这样就实现了更有效的

垃圾收集，也就减少了写操作和写入放大系数，同时获得了更高的写吞吐量，延长了驱动器的使用寿命。

Trim 命令和机制虽然看起来很美好，但是实际中会产生一些问题。原因在于，不同的 SSD 厂商对 Trim 命令的处理方式，以及具体的垃圾回收机制很不一样；有的实现还不错，有的就差强人意了，因此 Trim 的性能在每种 SSD 那里会有所不同。我们后面会提到，有些 SSD 的厂商的某些 SSD，因为对 Trim 的支持不太好，会造成某些情况下性能非常差。

还要注意的是，默认情况下，操作系统一般不启用 Trim。因此，当文件系统删除文件时，它只是将数据块标记为“未使用”。但是 SSD 控制器并不知道设备上的哪些页面可用，因此无法真正释放设备上的无效空间。所以，在没有启动 Trim 的情况下，一旦 SSD 设备的可用容量填满，即使文件系统知道设备上有可用容量，SSD 也会认为它自己已经存满。

那么怎么启动 Trim 呢？要在 SSD 上启用连续 Trim，必须在 mount 这块 SSD 的时候使用“Discard”安装选项。如果一块 SSD 已经安装了，想启动 Trim，那就需要卸载后重新安装，“Discard”选项才能生效。也就是说，使用 remount 命令是不起作用的。

所以，对于单个系统而言，最好在 grub 中启用 mount 选项，并重新启动。

想减少 SSD 损耗，却导致访问延迟过大?

Trim 的使用，虽然带来了**降低 SSD 损耗**的好处，但也带来了一些坏处，特别是 **IO 访问可能延迟加大**的问题。

为什么 Trim 会影响应用程序性能呢?

原因和 SSD 内部的实际机制有关。每个 SSD 内部都有一个 FTL（Flash Translation Layer）映射表，该表将操作系统的逻辑块地址（LBA，Logical Block Address）映射到 SSD 上的物理页面地址（PPA，Physical Page Address）。映射表在驱动器被写入时不断更新，以后每个读取和写入 IO 都要引用。

一般来说，映射表是存储在 SSD 驱动器的 RAM 中，以便快速访问；但是它的副本也存储在 SSD 中，目的是在电源故障时能够保留 LBA 到 PPA 的映射。随着 SSD 上面内容和数据的不断变化，这些变化包括新写入 IO 或垃圾回收，RAM 中的映射表也不断更新，并且持续写入 SSD 中。

如果在文件系统上启用了 Discard 选项，那么每次删除文件时，都会生成 Trim 命令。因为每次 Trim 都会更改映射表，所以对映射表的更改也就实际地记录到 SSD 中。这项操作可能需要花费比较长的时间，比如几毫秒的时间才能完成，在这个更改过程中，普通的数据读取和写入 I /O 会阻塞，并且阻塞到所有的映射表调整都被完全处理为止。当今业界的大多数 SSD 都是这样工作的。

上面我们看到，由于 Trim 的处理会阻塞普通的数据读取和写入 I /O，直到 Trim 完成映射表记录才返回，所以 Trim 的延迟对普通读写 I O 的延迟具有重大影响，尤其对高分位数（比如 P99、 P 99.9) 的读写 IO 延迟影响更大。减少 Trim 延迟就是减少 IO 延迟。所以，我们需要尽量减少 Trim 的等待处理时间。

另外值得你注意的是，每个 SSD 厂商和每款 SSD，对 Trim 的具体处理方式都可能不同，颇有些厂商的某些 SSD 具有严重的问题。我们生产实践中碰到过好几种这样的 SSD，比如有厂商的一种 SSD 在大量删除数据时有很大的延迟。这就要求我们在选购 SSD 时候，要特别小心，尤其是要做彻底的性能测试。

如何避免 Trim 带来的延迟?

我们刚才讲了用 Trim 的好处和坏处。好处是可以减少 SSD 的损耗，延长 SSD 的寿命；坏处是会造成应用程序的 IO 读写延迟变大。

那么怎么才能尽量避免 Trim 带来的坏处呢？我们这里谈两种方式：一是对 Discard 选项本身的调优，二是使用fstrim命令。这两种方式分别对应使用 Discard 被启用和不被启用的两种情况。

第一种方式是在启用了 Discard 后，对 Discard 的调优。对于已经启用 Discard 的场景下，Trim 命令默认是没有大小限制，也就是说，一次发送会尽可能多的删除命令。但是如果一次删除的数据太多，SSD 可能需要很长的时间才能返回，其他读写 IO 就会感受到很大的延迟。

那么我们就可以微调了，这里我们就可以借助另外一个参数，discard_max_bytes 对 Discard 进行调优。这个参数是一个操作系统内核参数，从名字也听得出，它可以指定一次 Trim 的最大数据量。

调整这个参数的优点，是可以根据实际可接受 IO 延迟的需要，来随意微调。举个例子，假如可接受 IO 延迟比较大，那就可以设置一个较大的 discard_max_byes 数值，比如 2GB。使用这个参数的坏处是，当有大文件删除时，如果没有相应的重新调整参数，Trim 的吞吐量会受影响。

第二种方式，是在没有启用 Discard 的场景下，采用 fstrim 来调优。fstrim 也是一个命令，它可以控制 Trim，来删除掉 SSD 上的文件系统不再使用的数据。默认情况下，fstrim 将删除文件系统中所有未使用的块，但是这个命令有其他的选项，根据删除范围或大小来进行微调。

这个命令一般用于 Discard 没有被启用的场景下。为了达到最好的效果，都是周期性的，或者采用外部事件触发来运行这个命令，比如用 Cron 来每天固定时间运行；或者每当 SSD 存储使用率到了某个大小就运行。

下图展示了一个实际生产环境中的性能数据。

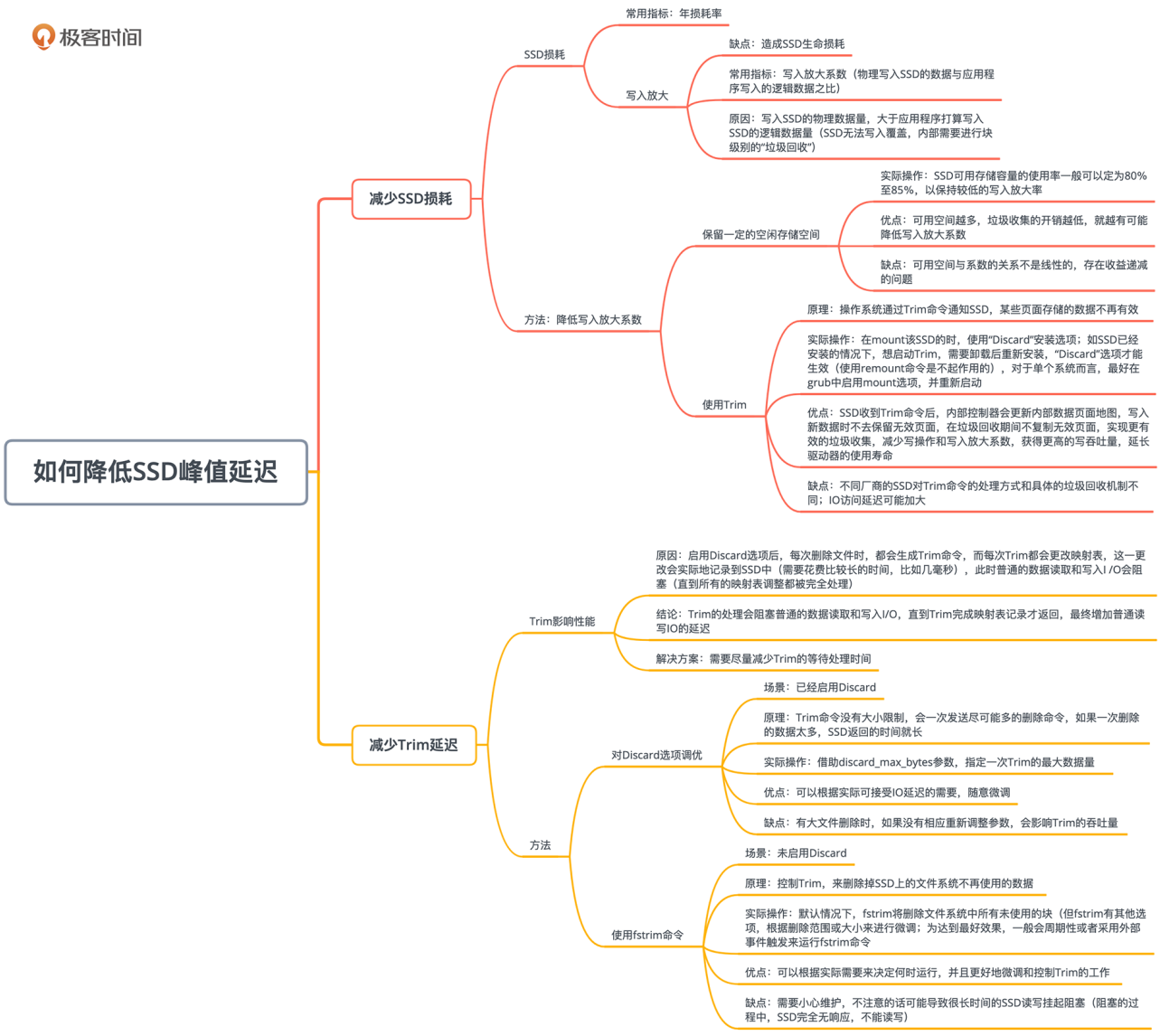
这是一个采用 fstrim 而降低 IO 延迟的例子。横轴是时间，纵轴是对 SSD 进行读操作的 IO 延迟。红色箭头是运行 fstrim 的时间。我们可以看到，在 fstrim 后，IO 延迟大幅度地降低了。

采用 fstrim 这个方式的优点是，可以根据实际需要来决定何时运行，并且更好地微调和控制 Trim 的工作。

这个方式也有缺点，就是如果不够小心，运行这个命令时可能导致很长时间的 SSD 读写挂起阻塞，在这个阻塞的过程中，SSD 完全没有响应，不能读写。我见过几次这样的生产例子，阻塞了好几分钟甚至几个小时的时间，整个 SSD 完全不能写入和读取数据。

总结

SSD 不断地重写会损坏其存储能力，就如同一口宝刀，不断地征战砍伐后，也会有缺口。这让我想起了唐代诗人马戴写的一首气势磅礴的《出塞词》：“金带连环束战袍，马头冲雪度临洮。卷旗夜劫单于帐，乱斫胡兵缺宝刀。”



为了延长 SSD 的寿命，我们可以采用 Trim 方式，以去除不必要的内部重写。

但是这种方式在某些特殊情况下，会增大外部 IO 的访问延迟。解决这一问题的方法是对 Trim 进行调优。我们这一讲就集中探讨了几种调优解决方案，来解决这一特殊情况下的问

题。

思考题

你们公司部署 SSD 了吗？有没有遇到关于损耗过大的问题，以及 Trim 的问题和相关讨论？最后采取的解决方案是什么呢？

欢迎你在留言区分享自己的思考，与我和其他同学一起讨论，也欢迎你把文章分享给自己的朋友。

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇　22 | 系统案例：如何提高iTLB（指令地址映射）的命中率？

下一篇　24 | 跨层案例：如何优化程序、OS和存储系统的交互？

精选留言 (2)

写留言

我来也
2020-01-19

现在很多都是用的云服务器了，不知道云厂商是怎么来调优的。

我们在使用云服务器的过程中，如果遇到了ssd的间歇性性能问题，有什么好办法来判断是自身程序的问题还是云服务器的问题呢？

之前在使用某云时，遇到过写入一个3M的日志，耗时2分多钟的情况，整个进程就租塞...

展开

作者回复: 这个问题的确诊断比较复杂，一个直观的方案就是对比。
比如如果其他服务用同样的云服务器，都好好的，那多半就是自己这个服务导致的。倒也不一定就是服务的程序有“问题”，只是需要考虑硬件特点来调优一下。
如果没有其他对比服务，可以试着自己写一个测试服务，用同样的云服务器。然后调节测试服务的各个参数，看看能不能把问题重现出来。

黑崽
2020-01-19

fstrim挂起之后，有办法干预吗？

展开

作者回复: 应该不行。原理上讲，操作系统的命令传给了SSD；挂起也多半是因为SSD内部导致的，操作系统也不能做啥了。